Inverse Kinematics and Design of a Novel 6-DoF Handheld Robot Arm

Austin Gregg-Smith and Walterio W. Mayol-Cuevas

***Abstract*— We present a novel 6-DoF cable driven manipulator for handheld robotic tasks. Based on a coupled tendon approach, the arm is optimized to maximize movement speed and configuration space while reducing the total mass of the arm. We propose a space carving approach to design optimal link geometry maximizing structural strength and joint limits while minimizing link mass. The design improves on similar non-handheld tendon-driven manipulators and reduces the required number of actuators to one per DoF. As the manipulator has one redundant joint, we present a 5-DoF inverse kinematics solution for the end effector pose. The inverse kinematics is solved by splitting the 6-DoF problem into two coupled 3-DoF problems and merging their results. A method for gracefully degrading the output of the inverse kinematics is described for cases where the desired end effector pose is outside the configuration space. This is useful for settings where the user is in the control loop and can help the robot to get closer to the desired location. The design of the handheld robot is offered as open source. While our results and tools are aimed at handheld robotics, the design and approach is useful to non-handheld applications.**

I. Introduction and Related Work

Handheld robotics [1] aims to develop cooperative robots where tasks are divided between robot and user for both physical and cognitive aspects of the task. When designing a handheld robot, some of the most important performance factors are its responsiveness, speed and mass. The manipulator must be lightweight so that it is not tiring to hold and also fast so that the robot can correct and compensate for the human's motions. These two requirements mean that the arm should be designed to have a high power to weight ratio yet be safe to use around people. A further concern is that the robot's weight should be distributed in a way that makes it easy to hold for long periods of time. The robot should be balanced evenly so that there is no constant torque required to hold it steady during operation.

These requirements call for the design of a low mass and compliant manipulator. There are several examples of compliant arm designs and many based on cable driven actuation [2]. Yet we note that the novel requirements of handheld robotics results in different design considerations. Our design (figure 1) is inspired by the cable driven mini 3D CT-arm [3][4][5], but with modifications suited for handheld applications. Their 6 degree of freedom (6-DoF) arm is 2.4m long and designed to inspect nuclear reactors. It was optimized for a long arm length so that it could reach deep inside a reactor and the end effector movement speed was not of primary concern. The joint axis in their design alternated by 90 degrees at every joint so that the arm could be wrapped around a drum for storage.

Department of Computer Science, University of Bristol, UK
agreggsmith@gmail.com, wmayol@cs.bris.ac.uk

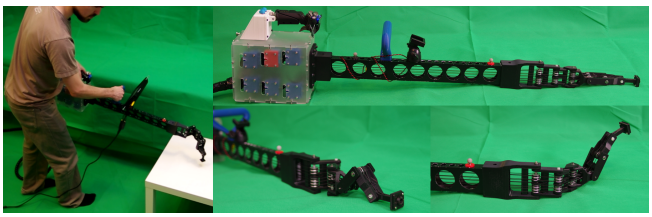


Fig. 1: Handheld 6-DoF robotic arm

The main advantage of their design is that all the motors are coupled. Usually when designing a manipulator the first joint requires the largest torque as it has the most resultant moments on it. The torque requirement for each subsequent joint is reduced as they don't have to carry as much mass. This makes finding appropriate actuators for each link difficult without over or under powering a particular joint. In a coupled design, all the motors work together to move the first joint. On the next joint, all the motors but one work together and so on. This means each joint has a summed torque proportional to the mass it has to lift and is therefore an optimal distribution of torque for the collective motor mass. A full analysis of these principles can be found in [5].

Another advantage of a cable driven design is that the motors are mounted in the base. This means the arm does not have to accelerate the mass of the motors during operation, allowing a larger payload per unit mass of motor. As the motors are in the base, they do not take up space on the active part of the manipulator which means it can be smaller.

The separation of the motors and manipulator helps meet the requirement of handheld robots because it gives flexibility when choosing the center of gravity of the arm. It is easy to move the motors by changing the length of the driving cables so the center of gravity is directly under where the user holds the robot. In contrast, a conventional serial robot arm places the motors along the length of the manipulator. This would create large torques about the base of the robot unless it was offset by an undesirable counterweight.

While reviewing common robotic actuation mechanisms, Whitney et al [6] notes that "Multi-link articulated cable drives running over low friction pulleys

and capstans offer perhaps the highest efficiency and smoothest operation among existing mechanical transmissions [7], [8]". All these factors combine to give an arm that is dexterous, fast and powerful for its size.

However, introducing a new arm design often results in the need to develop specific solutions to its inverse kinematics as generic solutions tend to be inefficient computationally and thus may limit the reaction and operational conditions that are critical in the agile and demanding handheld robot setting. This paper concentrates on both the design and the Inverse Kinematics of our handheld robot arm with section II describing the mechanical design of the arm including design features and link optimization via space carving while section III describes the coupled solution to the inverse kinematics.

II. Mechanical Design

When designing a cable driven actuator the first question is how to configure the joint axes to maximize the configuration space. The biggest constraint is that the wires from the joint closest to the end effector must pass through the rest of the arm with minimal friction whilst eliminating wire collisions. This constraint dominates the design process when attempting to maximize the individual joint limits.

While it is possible to have axial motion transmitted via a cable differential as in the WAM arm [8], the design is mechanically complex and does not fit well with a coupled design. For this reason we restricted the joint's axes to two directions. We iterated through possible configurations with these joint axis limits using physical prototypes before deciding on the configuration shown in figure 2. One advantage of this design is that the axes of adjacent joints are always parallel except between the third and fourth joint. This minimizes cable wear and friction as the driving tendons don't have to change direction more than strictly necessary. The configuration of two serially connected three link segments also allows for a inverse kinematics solution that takes advantage of the repeated structure as discussed in section III.

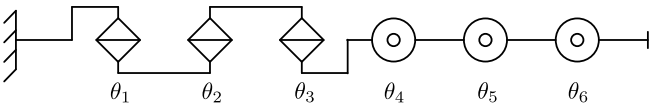


Fig. 2: Joint configuration

A. Design Features

1) Minimizing Actuator Requirements: The 2.4m long Mini 3D CT-Arm [5], uses 12 motors to control 6 degrees of freedom. They use pairs of motors to apply tension to each side of the wire for each of the 6 pulleys. Our design uses half the number of motors, so that one motor is directly linked to a corresponding pulley. The CT-arm uses potentiometers to measure the joint angles of each link but because the new arm has the motors directly coupled to each pulley, there is no longer a need for potentiometers as the relationship between motor angles and joint angles is fixed. This is in part possible because the arm is much smaller (30cm vs 240cm) so the elasticity of the tendons is less critical. These two improvements reduce weight and complexity significantly. Figure 3 shows the wiring configuration for all the joints and a model of the assembled arm. The green cylinders indicate motor driven pulleys, while the turquoise and blue cylinders show the link driving pulleys. The wires are shown wrapped around invisible idler pulleys to make the wiring clearer. The spacing between the wires has been expanded for clarity so the large angles the wires take between pulleys in the diagram are not representative of the built arm.

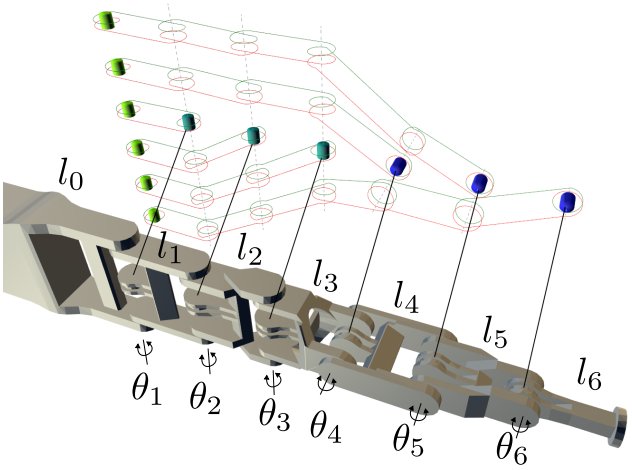


Fig. 3: Arm model and wiring diagram. The green cylinders indicate motor driven pulleys, while the turquoise and blue cylinders represent the link driving pulley. The base link is labeled l_0 and subsequent links increment by one to give l_1, l_2 etc. θ_1 is the angle between l_1 and l_0 and follows the same incrementing naming scheme as the links.

2) Link Design Via Space Carving: The arm joint configuration is comprised of a first section with three parallel joint axes, followed by a second section with three more joint axes rotated by 90 degrees relative to the first section. Figure 4 shows the configuration space of one of these three-link sections. The gray area is the space reachable by the beginning of link 3 (θ_3). The outer surface of the annular area is reached when $\theta_2 = 0$ and the inner edge when θ_2 is at the maximum allowed angle limit (θ_{2max}). The minimum possible distance from link 1 to link 3 is $\sin\left(\frac{\pi-\theta_2}{2}\right) 2L$. This distance is minimized when θ_2 is maximized so in order to have a large configuration space, it is important to maximize the joint limits of θ_2. In figure 4 right where $\theta_2 = \theta_{2max}$, in order to keep link 3 in the same axis as the base link, $\theta_1 = \theta_3 = -\frac{\theta_{2max}}{2}$, so maximizing these joint limits has a smaller effect on the size of the configuration space . It is still advantageous to maximize the other joint limits as much as possible, as they will increase the configuration

space, but they are of secondary priority.

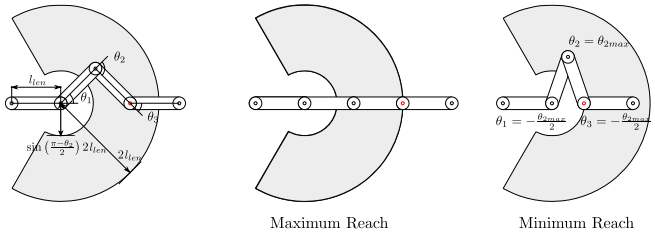
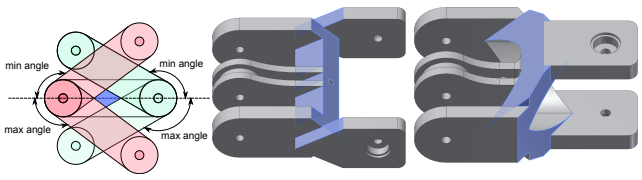


Fig. 4: Analysis of the workspace of a 3R arm used to optimize the joint limits

The same calculations apply to the second three link section for joint angles $\theta_4, \theta_5, \theta_6$. With this in mind, the links were designed to maximize the joint limits for θ_2 and θ_5 as a first priority and then to maximize the remaining limits where possible. The other criteria — minimizing size and mass while maximizing strength and rigidity — are at odds with maximizing joint limits because of the constraints large joint limits have on where material structure for the link can be placed.

To maximize both criteria, a form of space carving was used. When designing a link, the two connecting links are moved through their joint range to cut out the swept volume. The remaining material makes up the central core of the link as shown in blue in figure 5a. Figure 5b and 5c shows two CAD models of links produced using this technique. The middle shows link 3 which uses the same carving principle but has more complicated geometry due to the change of joint axis. An advantage of this technique is that adjacent links fit each other perfectly when they are at their maximum joint limits so the arm can be folded up into a compact and safe configuration during transportation.

Using this method the joint limits are as follows: $\theta_{1max} = \pm 135°$, $\theta_{2max} = \pm 150°$, $\theta_{3max} = \pm 135°$, $\theta_{4max} = \pm 125°$, $\theta_{5max} = \pm 150°$, $\theta_{6max} = \pm 150°$.



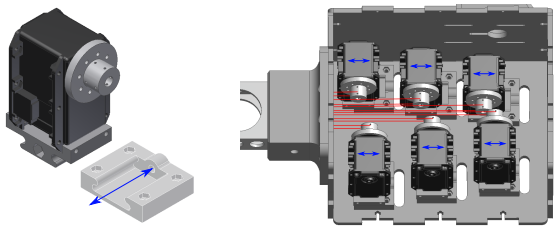
(a) (b) (c)

Fig. 5: Space carving method for minimizing link material. a): Material that is not carved away by rotating adjacent links is left in blue. b) and c): CAD models of link 1 and link 3 designed by using a space carving technique.

B. Hardware

The prototype arm links were 3D printed in ABS and the idler pulleys machined out of aluminum. The CAD files for all the components are open source and made available at [9]. The robot is powered by six Dynamixel MX-64T servos mounted on cable tensioning mechanisms. After the links and pulleys have been assembled, the driving cables are attached directly to each motor. The motors are mounted on a 3D printed saddle that fits into a corresponding bed shown disassembled in figure 6a. The motor is moved forward and backward in the bed by a bolt that acts as a lead screw. Figure 6b illustrates how the motors are arranged in a sideways "V" formation so that the motors can be individually tensioned without the cables crossing. When the arm is first turned on, it is calibrated by controller firmware recording the offset between the known zero angle of the arm, and the current angle recorded by the motor encoders. This calibration must be repeated every time the cables are de-tensioned or re-tensioned during maintenance.

The arm is used in an assisted inspection task where a user holds the robot and the actuated end effector autonomously seeks unexplored inspection zones [10]. The end effector can reach around obstacles with its tentacle-like design. The manipulator is also suitable for assisted pick and place, and/or painting tasks.



(a) CAD model of disassembled tensioner

(b) CAD model of the sideways "V" shaped servo layout with cables shown in red

Fig. 6: Cable tensioning mechanism and layout

III. 6-DoF Inverse Kinematics using coupled 3-DoF solutions

Although general purpose computational inverse kinematic (IK) solutions exist [11], [12] there are several advantages to an analytical solution. Although there have been performance improvements for computational methods, they are still generally based on algorithms that iteratively converge on a solution given some starting conditions and so are more computationally intensive than an analytical approach. Inverse kinematics problems often have multiple solutions but a computational algorithm will generally only return the solution that is nearest to the input starting solution. This is an undesirable characteristic when using the IK results as the input to a path planner because the solution that converges from the IK might not be the configuration that gives the best results from the path planner. An analytical solution will provide all the valid configurations that reach a pose which gives the path planner more options to find short paths to that pose.

Figure 7 shows the nomenclature used when solving the inverse kinematics problem. L is defined as the line

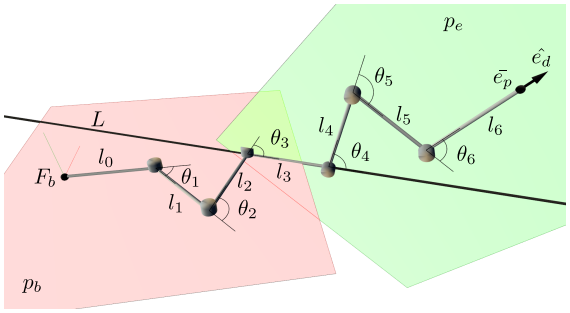


Fig. 7: Planes for solving each 3-DoF problem. The base plane is in red and the tip plane is in green. The inverse kinematics is solved in 2D on each plane for the corresponding section of 3 parallel joints. Their results are merged along the common line between the two planes where they intersect.

common to both the plane defined by the base of the robot (red) , and the plane defined by the desired end effector pose (green). The kinematics of the arm has one redundant joint so in most cases there are an infinite number of positions where l_3 can lie along L while l_0 and l_6 stay fixed. Practically, this means you can slide l_3 forward and backwards along L until l_3 collides with another link or reaches one of the joint limits. The main problem when solving the inverse kinematics is choosing the optimum position for l_3.

The arm is made up of two sets of joints that have two common axes of rotation. The first set is defined as all the joints and links that lie on the base plane p_b which are $B = \{\theta_1, \theta_2, \theta_3\}, \{l_0, l_1, l_2, l_3\}$ while the second set $E = \{\theta_4, \theta_5, \theta_6\}, \{l_3, l_4, l_5, l_6\}$ all lie on the end effector plane p_e shown in green. Note that l_3 is contained by both sets and lies on line L which is defined by where p_b and p_e intersect. The 6-DoF problem can be split into two 3-DoF problems defined by B and E that each lie on their own plane.

Problem B : Given the pose of l_0 and L, where can l_3 lie on L without any collisions between l_0, l_1, l_2, l_3 and keeping $\theta_1, \theta_2, \theta_3$ within their joint limits?

Problem E : Given the pose of l_6 and L, where can l_3 lie on L without any collisions between l_3, l_4, l_5, l_6 and keeping $\theta_4, \theta_5, \theta_6$ within their joint limits?

The solution to each problem is a list of possible locations that l_3 can occupy. When both B and E have been solved, the position of l_3 is chosen as all locations where the valid locations overlap. This ensures that all collision constraints and joint angle limits are met. Once the position of l_3 is known the problem is fully constrained and it is straightforward to solve for the joint angles $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$ using geometry.

B and E are essentially the same problem but with the direction and associated links reversed. We define a function:

$$l_3 = ik3D(l_{fixed}, l_a, l_b, l_{onLine}, L, \theta_{aMax}, \theta_{bMax}, \theta_{cMax})$$

Problem B is expressed as:

$$l_3 = ik3D(l_0, l_1, l_2, l_3, L, \theta_{1Max}, \theta_{2Max}, \theta_{3Max})$$

and problem E as:

$$l_3 = ik3D(l_6, l_5, l_4, l_3, L, \theta_{6Max}, \theta_{5Max}, \theta_{4Max})$$

Section III-A explains how to calculate the joint and link collision constraints. Section III-B shows how to enumerate possible locations for l_3 given the collision constraints and section III-C merges the results of B and E into a 6-DoF solution.

A. Calculating Valid 3-DoF Solution Boundaries

The black line with the arrow in figures (8-14) is the a "line query" which is 2D projection of L and indicates the direction and range of positions l_3 must lie on. Figure 8 illustrates the links and their associated joint limits which are color coded the same way across all the diagrams mentioned above. In this section, the geometry will be discussed in terms of problem B with $\theta_1, \theta_2, \theta_3, l_0, l_1, l_2, l_3$, but the same logic applies to problem E where l_0 is replaced by l_6, $l_1$by l_5, etc and θ_1 by θ_6, θ_2 by θ_5, etc.

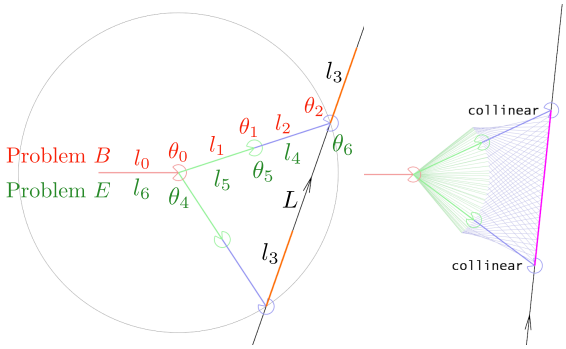


Fig. 8: Both link's are collinear boundary

If joint limits and link-to-link collisions are ignored, the inverse kinematics solution for the 3 joint section is trivial and well known [13]. In figure 8, l_3 would be able to reach any location inside the gray circle and therefore any point where L is inside the circle. If the angle and collision constraints are added this is no longer guaranteed to be true and more analysis must be done.

Figure 8 shows the simplest example of a boundary condition. The two boundaries of where l_3 can lie are at the intersection points of the gray circle and L. Figure 8 right shows all the places that l_3 can lie on L in magenta. The link configurations to reach the magenta points are drawn in a lighter color for clarity.

In the following section we describe how to calculate the boundary solutions for seven different joint and collision constraints. Once all the boundary conditions have been described, we will explain how to combine their results.

The seven boundary constraints are:

1) l_1 and l_2 are collinear: 2 solutions (figure 8)

2) $\theta_1 = \theta_{1max}$: 4 solutions (figure 9)
3) $\theta_2 = \theta_{2max}$: 4 solutions (figure 10)
4) $\theta_3 = \theta_{3max}$: 4 solutions (figure 11)
5) l_2 collides with l_0: 2 solutions (figure 12)
6) l_3 collides with l_0: 2 solutions (figure 13)
7) l_3 collides with l_1: 1 solution (figure 14)

1) Links 1 and 2 are Collinear: The first and simplest case is where the l_1 and l_2 lines are collinear as shown by figure 8. The two possible solutions are found at the intersection between L and a circle with $radius = 2l_{len}$. The third link is shown in orange and is collinear with L and points in the same direction. In the subsequent examples l_3 is not always shown because it often makes the diagram harder to understand, but it should be assumed to be there.

2) Joint 1 Constraints: Figure 9 right shows four possible solutions where θ_1 is at its maximum range of travel. l_1 can have two possible locations when $\theta_1 = \pm\theta_{1Max}$. For each of these two locations, the valid locations for l_3 are the intersection point of the L and the circle defined by the $radius = l_{len}$ as shown by figure 9 left. In this example, two of the four solutions have been rejected because they are past the joint limits of θ_3 and are draw in gray.

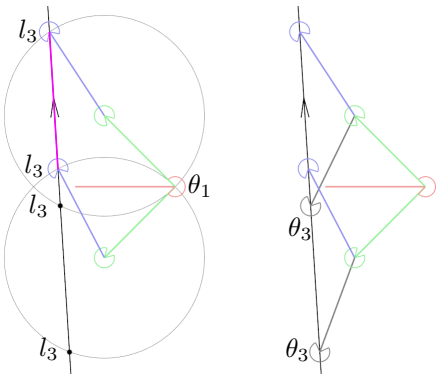


Fig. 9: Joint 1 at maximum limit boundary

3) Joint 2 Constraints: Figure 10 left shows the construction method for calculating the solutions where θ_2 is at it's maximum range. A triangle ABC is constructed with two sides of $length = l_{len}$ and angle ABC set so that θ_2 is at its maximum angle. The length of the third side defines the radius R of the circle (black) that all the valid solutions must lie on. The positions of l_3 are the points where the circle intersects L. Figure 10 right shows the same configuration, but shows all four candidates including the two invalid solutions (gray) that are rejected because they exceed the joint limits of θ_1 or θ_3. Note that the reachable areas are the regions of L that are outside the black circle shown in magenta because inside the circle is the region that breaks the constraints of θ_2. The same is true for the other boundary conditions but it is difficult to show graphically so they are omitted from the diagrams.

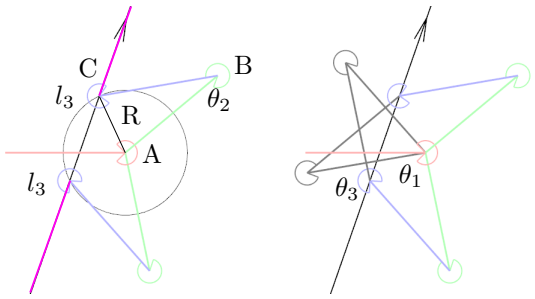


Fig. 10: Joint 2 at maximum limit boundary

4) Joint 3 Constraints: Figure 11 right shows four possible solutions where θ_3 is at its maximum angle relative to L. In figure 11 left the angle LDE is set from the maximum θ_3 angle and is used to calculate length N. Lines L_a, L_b are offset from L by length N. The intersection points between L_a and L_b and circle C are the candidate positions for θ_2. Figure 11 right shows the two solutions shown in gray that were rejected for exceeding the maximum angle of θ_2.

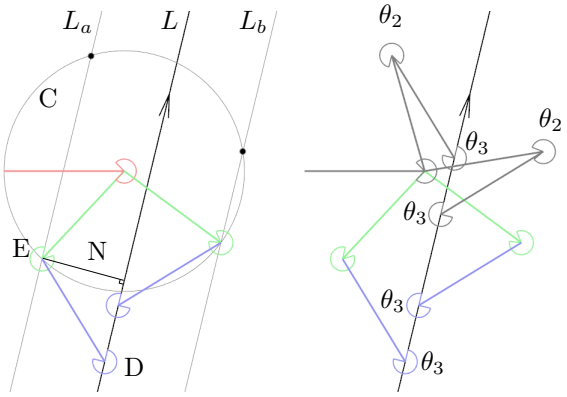


Fig. 11: Solutions where joint 3 is at its maximum limit boundary

5) Link 2 Collides with Link 0: Figure 12 left shows the case where l_2 collides with l_0. l_3 must lie on the intersection point of L and l_0. The possible positions for θ_2 are the intersections between the blue circle and the green circle. Figure 12 right shows all possible solutions including the grayed out one rejected for exceeding θ_3 angle limits.

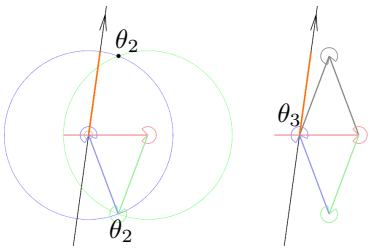


Fig. 12: Link 2 collides with link 0

6) Link 3 Collides with Link 0: Figure 13 shows the case where l_3 collides with l_0. The intersection of L and l_0 is found. l_3 must then be 1 link length backward along L from the intersection point. A blue circle is

constructed around this point and intersected with the green circle around l_0 to find the two candidates for θ_2. Figure 13 right shows a valid solution and an invalid solution that exceeds θ_1 angle limits.

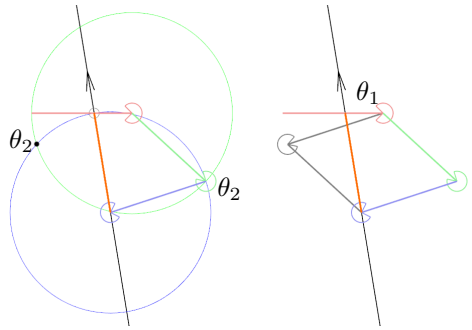


Fig. 13: Link 3 collides with link 0

7) Link 3 Collides with Link 1: The last case cannot be solved using geometric constraints alone. Given the constraints that the end of l_3 must touch l_1 and also lie on L, there is no closed form solution for the joint angles that meets these constraints, as shown in figure 14 right. In order to calculate the position of any link, the position of one of the other links must be known already. For example, the angle of θ_1 cannot be known without knowing how far along L l_3 lies, but the position of l_3 is dependent on the angle of θ_1.

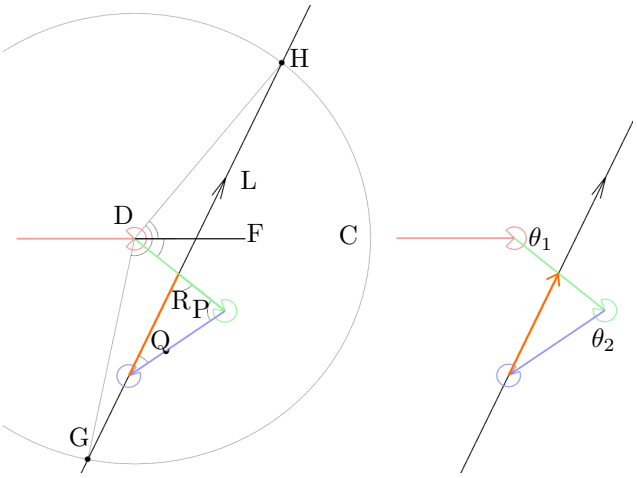


Fig. 14: Link 3 collides with link 1

Instead an interpolation method was used to calculate the link configuration that meets all these constraints as shown in figure 14. A list of valid solutions are generated by calculating the forward kinematics of a series of joint angles that match the constraints. When looking at this problem from a forward kinematics perspective, the aim is to produce an isosceles triangle PQR, where angle RPQ and angle QRP have the same magnitude. To generate this set of valid isosceles triangles the following joint angles are chosen:

$$\theta_1 = 0, \quad \theta_2 \in [j_{3Max}/2, \frac{2}{3}\pi], \quad \theta_3 = 2(\pi - \theta_2) \qquad (1)$$

Once all the link positions have been calculated a line L is constructed to lie on l_3. Iterating through the forward kinematics gives us a mapping of joint angles to the line L. To solve the inverse problem, we want to know the joint angles given an arbitrary line L. The problem is constrained enough that if we can calculate the mapping from L to the angle of θ_1, θ_2 or θ_3, we can work out the rest of the geometry through normal geometric constraints.

In this example, we chose to solve for θ_1 as a function of L, where L is parametrized as a 2D position and direction:

$$\theta_1 = f(L) = f(p_x, p_y, d_x, d_y) \qquad (2)$$

At first glance it appears that the interpolation function will require four inputs to describe the directed line, however the problem can be constrained to 2-DoF. The line L is constrained so that it must touch the circle C at two points (G,H) so the line can be parametrized by the angle GDH and FDH.

$$\theta_1 = f(GDH, FDH) \qquad (3)$$

This is still 2-DoF and for simplicity it would be better to only interpolate over a 1D function instead of 2D. If the angle GDH stays constant and angle FDH changes:

$$\theta_1 = PDH - FDH \qquad (4)$$

To solve for the angle of θ_1 using PDH, we create an interpolation function from equations (2), (3), and (4) that accepts angle GDH and returns PDH.

$$PDH = f_1(GDH) \qquad (5)$$

Combining equations (3), (4), and (5) gives:

$$\theta_1 = f(GDH, FDH) = f_1(GDH) - FDH \qquad (6)$$

Once θ_1 angle is known, the position θ_2 can be calculated using trigonometry. The position of l_3 is the intersection of a circle around θ_2 and the line L. The problem is then fully solved.

B. Calculating Valid 3-DoF Solution Ranges

The previous section covered the seven behaviors that mark the boundaries of sets of solutions given different constraints. In some cases, all the solutions on one side of the boundary solution will be valid and on the other side will be invalid. The other case is where the solutions will be valid on both sides of a boundary. Regardless, the regions of valid solutions are always enclosed by a pair of solutions calculated above. In order to calculate the range of valid solutions, the spaces between the calculated boundaries are tested and classified as valid or invalid.

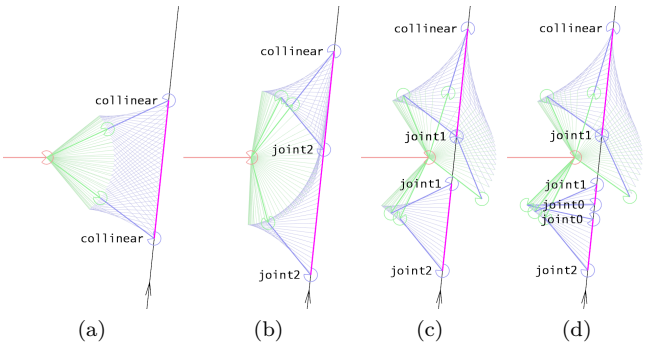


Fig. 15: A list of L queries and the corresponding valid solution ranges

Figure 15 shows valid ranges of solutions for a number of desired line queries. Part (a) shows the simplest case; where the boundary solutions are both straight lines. The valid solutions between the boundary cases are shown in a lighter color and the point where they meet the desired line is highlighted in magenta. As the desired line moves to the left, part (b) shows that the top boundary condition remains "collinear" but the bottom "collinear" has been replaced by "joint2" because given the current L, the collinear solution is no longer valid and has been replaced by a "joint2" solution. An extra "joint2" solution is also now valid in the middle of the line but since there are valid solutions to either side of the "joint2" it does not mark a reachability boundary. Part (c) shows that as L moves further to the left, the once continuous magenta region is now split into two separate segments. Starting from the bottom of (c), the first segment is surround by the "joint2" and "joint1" boundary conditions while the second segment is contained by "joint1" and "collinear" boundaries. The region of L between these two segments is not reachable as at least one of the seven constraints are not met. Section (d) shows the first valid region between "joint2" and "joint1" get split by a pair of "joint0" boundaries, while the second valid region from (c) remains unchanged. The process of calculating the valid regions is the same for any desired line query.

C. Merging Valid 3-DoF Solution Ranges

Up until this point, we have only been considering the solution of a 3-DoF problem. As described in section III-A the 6-DoF problem can be considered as consisting of two 3-DoF problems taking place in two different orthogonal planes.

Figure 16 shows the output of two sets of two 3-DoF problems and their overlapping sections. The 3-DoF problem has to be solved four times because direction of l_3 is important for the constraints. First l_3 is chosen to face to the right while lying on L and the 3-DoF problem is solved once for the base plane.

Next we draw the magenta region described in the previous section as red rectangles with arrows facing to the right on the far side of L. The same is done for the end effector plane, whose valid regions are drawn as

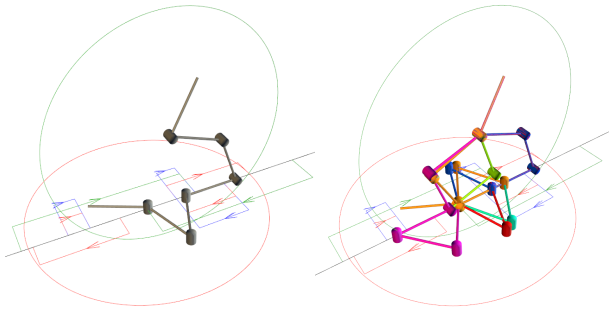


Fig. 16: Base (red) and tip (green) valid solution ranges are shown as boxes with arrows indicate the direction of the common link. Blue boxes show the overlap between the base and tip ranges. Right: One valid solution. Left: All valid solutions

green rectangles with arrows facing to the right. Then we reverse the direction of l_3 and solve the problem again for the base and end effector planes. The results are shown as green and red rectangles on the near side of L with arrows facing to the left.

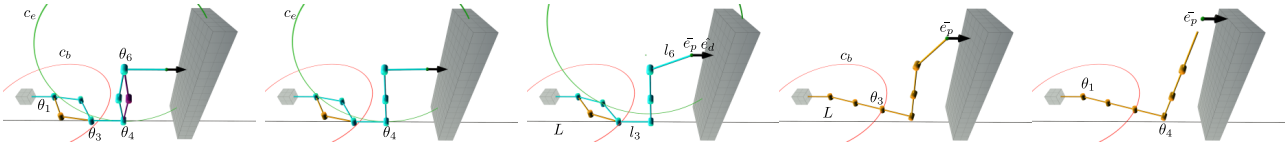
At this point, all of the constraints for both the base and end effector sections are described in a common reference frame and they can now be merged to find a global solution. The blue rectangles are regions of overlap between the valid base ranges (red) and valid end effector ranges (green). l_3 must lie somewhere inside the blue range to meet the combined constraints of both the base and tip sections.

In figure 16 left the solution shown in gray is only one of many possible solutions. Each of the calculated blue ranges has an infinite number of solutions so only the midpoint of the range is chosen as the output of the function. The midpoint is chosen rather than either end of the range because it distributes the magnitude of the joint angles more evenly across all the joints. Figure 16 right shows the same configuration as figure 16 left but with all valid midpoints of the blue solutions plotted. In this case there exist six solutions, but it is possible to have up to the (number of ranges) $\times$(number of directions) $\times$(number of base solutions) $\times$(number of end effector solutions) $= 3 \times 2 \times 2 \times 2 = 24$ solutions.

When operating in an unconstrained environment, it is often sufficient to choose the solution that is closest to the current configuration of the arm. However, if there are obstacles that the robot must avoid, some of the calculated solutions will be invalid due to the links colliding with the environment. In this case a path planning algorithm can use all the possible solutions and choose the one that minimizes collision.

D. Solution's Graceful Degradation

The above sections describe the solution to the 5-DoF inverse kinematics problem. However there are cases where the arm does not have a 5-DoF solution to a desired pose. In these cases it is desirable that the output of the inverse kinematics function should return the



(a) 5-DoF (b) 5-DoF boundary (c) 4-DoF (d) 3-DoF (e) 2-DoF

Fig. 17: Graceful degradation offers as IK output the closest pose to guide the user and implicitly prompts for help in reaching the goal

closest possible pose. We call this graceful degradation and it is achieved by reducing the number of DoF that the end effector attempts to reach as illustrated in figure 17. From the user's perspective of holding the robot as a handheld tool, this is intuitive as the reduced solution can be interpreted as a pointing gesture by the robot, to indicate where the user should move the robot towards.

The desired end effector pose is a position $\bar{e_p}$ and direction $\hat{e_d}$ and drawn as a black arrow in figure 17. Figure 17a shows the arm reaching the desired pose successfully. Note that θ_3 is inside the red base circle c_b defined about the center of θ_1 with $radius = 2l_{len}$ and θ_4 is inside the end effector circle c_e centered about θ_6 shown in green. In this case, there are two ways for the base links to reach l_3 (shown in cyan and orange), and two ways for the end effector links to reach to l_3 (shown in cyan and purple). In (b) the desired pose has moved up on the gray cuboid and θ_4 is now touching the edge of c_e. This causes the two end effector solutions shown in (a) as cyan and purple to collapse to the one shown in cyan in (b). This is the boundary between 5-DoF and 4-DoF solutions, because if the desired pose moves farther up the cube there are no more valid 5-DoF solutions and only 4-DoF can be reached, as illustrated in (c).

The desired end effector direction $\hat{e_d}$ can be parametrized by pitch and yaw angles defined relative to the base. Once the end effector pose is too far away to reach with 5-DoF it is still possible to reach $\bar{e_p}$ if the desired pitch or yaw constraints of $\hat{e_d}$ are relaxed. In figure 17c, l_6 is not longer collinear with $\hat{e_d}$ because the pitch constraint is not met but the yaw constraint is still met because l_3 still lies on L.

In figure 17d l_3 does not lie on L any more so both the pitch and yaw constrains are not met and only the 3-DoF $\bar{e_p}$ has been reached. Note that in (c) there are two base plane solutions (shown in orange and cyan) that collapse down to one solution in (d) where θ_3 cannot lie on the intersection of c_b and L, so it now lies on the edge c_b that is closest to $\bar{e_p}$.

If $\bar{e_p}$ moves further up the gray box it moves entirely out of range of the arm so that only θ_1 and θ_4 are required to move the end effector as close as possible to $\bar{e_p}$ resulting in a 2-DoF solution. In figure 17, the pitch constraint is relaxed before the yaw constraint because $\bar{e_p}$ moves such that θ_4 hits the edge of c_e before θ_3 hits the edge of c_b. However if $\bar{e_p}$ is moved towards us, the yaw constraint will need to be relaxed before the pitch using the same principles outlined above.

IV. Conclusions

We present a novel design optimized for the nascent area of handheld robotic applications through the use of link to link space carving. The inverse kinematics are solved analytically by splitting the 6-DoF problem into two coupled 3-DoF problems and merging the results. A method for gracefully degrading the number of DoF that the arm reaches when the desired end effector pose is outside the arm's configuration space is also proposed. The arm design and CAD files are released as open source hardware at [9].

References

[1] A. Gregg-Smith and W. W. Mayol-Cuevas, "The Design and Evaluation of a Cooperative Handheld Robot," in *ICRA*, 2015.

[2] R. J. Webster and B. A. Jones, "Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review," *The International Journal of Robotics Research*, vol. 29, no. 13, 2010.

[3] S. Ma, S. Hirose, and H. Yoshinada, "Design and experiments for a coupled tendon-driven manipulator," *IEEE Control Systems*, vol. 13, no. 1, feb 1993.

[4] S. Hirose and S. Ma, "Coupled tendon-driven multijoint manipulator," in *ICRA 1991*, 1991.

[5] A. Horigome, H. Yamada, G. Endo, S. Shin, S. Hirose, and E. F. Fukushima, "Development of a Coupled Tendon-Driven 3D Multi-Joint Manipulator," in *ICRA*, 2014.

[6] J. Whitney, M. Glisson, E. Brockmeyer, and J. Hodgins, "A Low-Friction Passive Fluid Transmission and Fluid-Tendon Soft Actuator," in *IROS*, 2014.

[7] S. Charles, H. Das, T. Ohm, C. Boswell, G. Rodriguez, R. Steele, and D. Istrate, "Dexterity-enhanced telerobotic microsurgery," in *ICAR*, 1997.

[8] Townsend W and Salisbury J, "Mechanical design for whole-arm manipulation," in *Robots and Biological Systems: Towards a New Bionics?* Springer, 1993, pp. 153–164.

[9] "http://www.handheldrobotics.org."

[10] A. Gregg-Smith and W. W. Mayol-Cuevas, "Investigating Spatial Guidance For a Cooperative Handheld Robot," in *ICRA*, 2016.

[11] A. Aristidou and J. Lasenby, "FABRIK: A fast, iterative solver for the Inverse Kinematics problem," *Graphical Models*, vol. 73, no. 5, sep 2011.

[12] R. Mukundan, "A Fast Inverse Kinematics Solution for an n-link Joint Chain," *ICITA*, 2008.

[13] J. J. Craig, *Introduction to Robotics Mechanics and Control*. Prentice Hall, 1985.